AMENDMENTS TO THE SPECIFICATION:

Page 1, 2nd paragraph:

Web services are a form of distributed computing, in which one device (a client)

calls procedures provided on another device (server) so as to use the services provided by

that server. There are a number of different distributed computing applications in which

various different protocols are used such as CORBA and DCOM. Distributed systems

may use a variety of different means for the client to call the procedure on the server,

such as remote method ~~in vocation~~ <u>invocation</u> (RMI), remote procedure calling (RPC) or

message queuing.

Page 1, 3rd paragraph:

Web services can be considered as a collection of functions which have been

packaged together and published to a network for use by clients within the network.

They provide the building blocks for creating open distributed systems, and as such any

number of Web services can be combined to form more complicated, higher level

service. Today, Web services are used to enable communication between computers in

the form of messaging and RPC mechanisms across IP networks. Essentially, the

advantages of Web services over other distributed computing arrangements are that they

are particularly suited for heterogeneous environments such as the internet. The reason

for this is that the Web services use an XML-based communication protocol which is

light weight and easily understandable by all of the various different Web services. In

addition, the Web services operate by transmitting communication messages using any

underlying network communication protocols, but in particular use HTTP which is

ubiquitous throughout the internet. The advantages of Web services in the use of HTTP

transport and XML encoding [[which]] are supported by many computing platforms such

as JAVA and MICROSOFT. One example of a Web service is MICROSOFT

PASSPORT (an authentication service hosted by MICROSOFT).

Pages 1-2, bridging paragraph:

The protocol stack for Web services comprises, at the top, the Web services

applications which are offered by service providers for access by a service requester

(client). Under this, the XML-based communication mechanism mentioned earlier is

typically SOAP (Simple Object Access Protocol)--this XML-based standard is a

messaging framework designed for exchanging structured information in a distributed

environment over a variety of underlying protocols, but is lightweight in that it misses out

many advanced features such as reliability, security, and routing. The ~~XML-bassed~~

1305975

XML-based messaging protocol operates over the underlying network communication

protocols ([[eg]] e.g., HTTP). These features of Web services mean that they provide one

of the best interfaces for interoperability between legacy systems, [[Java]] JAVA and

[[.Net]] .NET systems. Unfortunately, they do suffer from some limitations, in particular

load balancing and load sharing cannot be supported in the normal way.

Page 2, 1st full paragraph:

Earlier BRITISH TELECOMMUNICATIONS patent application

PCT/GB02/03981 is directed towards a system which overcomes some of the limitations

encountered in distributed computing. In particular, the system addresses the problems

which can arise between a client-sever relationship when one or more clients overuse the

capabilities of the servers, and solve these using the compulsory download of a client side

intermediary which acts to control the call rates allowed to the server. This thereby

prevents the server from overuse by throttling back the call rate in the event that the

server becomes congested, and offering better load control of the services offered by the

servers. However, this system is directed towards a single client-sever relationship, and

as such does not address the problems encountered in a multi-sever environment of high

availability Web services, in which duplicate Web services are operated on several

different servers. In particular, the failover capability should one of the servers or Web services fail is not addressed.

Pages 3-4, bridging paragraph:

This procedure of "binding", linking the WSDL to URL and then to IP address, is maintained throughout the lifecycle of the client, unless the client specifically demands a re-bind. In this case, all further calls to the service are performed without reference again to the WSDL. If the server becomes congested or fails then the client only notices when it tries to send a SOAP message to the server and the process eventually fails. In this case, if the WSDL has multiple service names and ports specified, then the client can attempt to rebind to another one. However, even if achieved this will have caused a disruption to the service offered to the client. In addition, if the client has not been programmed to cater for such a condition, then the client will fail. In addition to the problems encountered during failure of a server, no distribution of loading is carried out since the client will only send SOAP requests to one server (service port) unless the client is forced to rebind before it sends every message. However, such dynamic rebinding would require special programming by the client and in some cases the Web services SDK (software development kit) supplied with .NET or JAVA may not support [[It]] it.

Patrick B. FARLEY, *et al.*
Serial No. 10/549,358
November 13, 2008

In some cases, Web service bindings may last for longer than the planned SOAP server

uptime, thus when a server is taken down for maintenance the client will suffer failed

responses.